## 22.3 A 160kGate 4.5kB SRAM H.264 Video Decoder for HDTV Applications

C. C. Lin, J. I. Guo, H. C. Chang, Y. C. Yang, J. W. Chen, M. C. Tsai, J. S. Wang

National Chung-Cheng University, Chia-Yi, Taiwan

Observing the design trends in the evolution of multimedia ASIC designs during the past decade, one finds that efficient MPEG-2, MPEG-4, JPEG-2000 encoder/decoder/CODEC designs have been designed with increasing hardware efficiency and lower power consumption. With respect to implementations of MPEG-4 AVC/H.264 [1], there has been one single-chip video encoder [2] and some video decoder cores [3-6] proposed since 2004. This paper presents a low-cost H.264 video decoder for HDTV applications. Compared to a state-of-the-art H.264 video decoder [3], the proposed design can reduce the gate-count by 46% and internal memory by 93% for real-time HD1080 video decoding.

Figure 22.3.1 shows the block diagram of the design with a hybrid block-level pipelined architecture. To reduce hardware complexity, the following major functional blocks are optimized: the 4×4 block-level pipelined Bit-Stream Decoder (BSD, including Parser, CAVLD, CABAD, and IDS), IQ/IT, and ILF, as well as a hybrid block-level pipelined PPC. To reduce internal memory, a DMA-like PDSB is used for quickly accessing the correlated data in decoding MBs from external memories. To simplify decoding of headers above the slice layer, a system controller is used with an external RISC. To reduce operating frequency in external memory interfaces, dual external memories are adopted with separate AHB buses. Using TSMC 1P6M 0.18μm CMOS technology, the design achieves real-time H.264 decoding on HD1080 video when operating at 120MHz with 320mW power consumption at the cost of 160k gates and 4.5kB of internal memory. Highlights of key techniques utilized in the design follow.

The first challenge, is the eliminatation of the bottleneck in the CABAD where decoding one bit of codeword needs 3 cycles, which is caused by the dependency of renormalizing the probability model iteratively. The context memory is partitioned into 6 segmented context memories accompanied with a pipeline schedule for reducing the processing latency by 33%, as illustrated in Fig. 22.3.2. Then, context cache registers are added to the segmented context memory for loading successive contexts in a cycle with a look-ahead codeword parsing scheme to decode two bits of codeword in a cycle by exploiting the fact that no additional cycle is needed to renormalize the probability model of bit-streams if the look-ahead condition specified in Fig. 22.3.2 fits. Combining these techniques can contribute to a 57% cycle count reduction in CABAD.

The second challenge is to optimize the PPC operations from two perspectives. One is simplifying all the prediction modes by sharing common operations and hardware. The other is reducing the data bandwidth needed for compensation. As illustrated in Fig. 22.3.3, regularity exists in all the prediction modes after exploiting specific I/O data reordering. This regularity implies three facts. One is that some output samples are identical (e.g., samples $b$ and $e$ are the same), which benefits the complexity reduction. Another fact is that some common terms can be shared in computing different output samples (e.g., term $(B+C)$ can be shared in computing $a$ and $b$). The other fact is that computing different output samples can be viewed as doing the same filtering operations on shifted input samples, which benefits carrying out the filter operations by the same hardware. Exploiting this regularity can contribute to a 60% complexity reduction in PPC.

In addition, variable block size motion compensation (MC) is used that finds a good trade-off in increasing data re-use and reducing local memory size. For doing MC operations on data blocks with 1/4-pel motion vectors, the realization in H.264 reference software (JM) fetches reference data in units of 9×9 for each 4×4 block without eliminating the redundant data access among neighboring blocks, as illustrated in Fig. 22.3.4. Instead, the proposed MC scheme provides the flexibility to respectively fetch reference data in units of 13×13, 13×9, 9×13, and 9×9 for data encoded by 8×8, 8×4, 4×8, and 4×4 blocks, which eliminates overlapped accesses and contributes to a 48% reduction in data bandwidth. In summary, the decoding process of H.264 video is optimized through various design techniques with the performance results summarized in Fig. 22.3.4. These techniques altogether contribute to a 46% reduction in gate-count as compared to the design in [3].

In addition to complexity reduction, there was a focus on reducing the large internal memory required in previous designs [3,4,5], which is caused by storing large amounts of correlated data during decoding the current MB and its neighboring MBs. If all the correlated data are stored in internal memory, about 26kB of internal memory is required for supporting HD1080 video decoding. Therefore, a DMA-like PDSB is proposed, as shown in Fig. 22.3.5, which collects the correlated data in a MB and stores them to external memory if they are not used immediately. Using this technique, the internal memory can be reduced from 26kB to only 0.5kB, which results in 98% reduction in memory size with less than 10% increase in external memory bandwidth. Finally, dual external memories are adopted to reduce the operating frequency in the memory interfaces. According to cycle-accurate simulation of the proposed design on an ARM-based platform, a 120MHz clock can afford enough memory bandwidth for HD1080 video decoding.

Fig. 22.3.6 summarizes the chip implementation. The core size is 2.9×2.9 $mm^2$ with a micrograph shown in Fig. 22.3.7. The hardware includes 160k gates and 4.5kB of internal memory. The power consumption is 320mW at 120MHz for real-time decoding of HD1080 video. Based on a 0.13μm CMOS technology with 1.2V supply, the power consumption of the design is 108mW at 120MHz. As shown in Fig. 22.3.6, the design outperforms the state-of-the-art H.264 decoder [3] in terms of a 46% reduction in gate-count and a 93% reduction in internal memory.

*Acknowledgements:*
The authors thank the National Science Council, the Chip Implementation Center (CIC), and Faraday Inc. of Taiwan for granting the research, supporting chip fabrication, and providing FPGA environment, respectively.

*References:*

[1] Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG, "ISO/IEC 14496-10," May, 2003.
[2] Y. W. Huang, et al., "A 1.3TOPS H.264/AVC Single-Chip Encoder for HDTV Applications," *ISSCC Dig. Tech. Papers*, pp. 128-129, Feb., 2005.
[3] Y. Hu, A. Simpson, K. McAdoo, and J. Cush, "A High Definition H.264/AVC Hardware Video Decoder Core for Multimedia SoC's," *Proc. ISCE 2004*, pp. 385-389, Sept., 2004.
[4] T. W. Chen, et al., "Architecture Design of H.264/AVC Decoder with Hybrid Task Pipelining for High Definition Videos," *Proc. ISCAS*, pp. 2931-2934, May, 2005.
[5] T. A. Lin, et al., "An H.264/AVC Decoder with 4×4 Block-Level Pipeline," *Proc. ISCAS*, pp. 1810-1813, May, 2005.
[6] H. Y. Kang, et al., "MPEG-4 AVC/H.264 Decoder with Scalable Bus Architecture and Dual Memory Controller," *Proc. ISCAS*, vol. II, pp. 145-148, May, 2004.

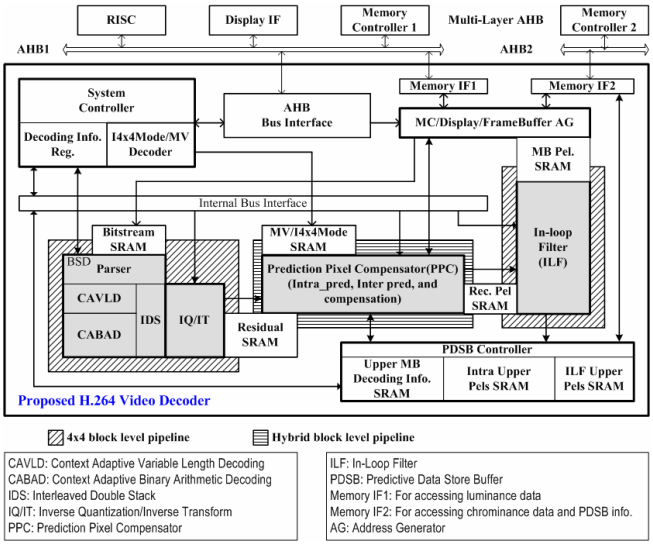



**Figure 22.3.1: Block diagram of the proposed H.264 video decoder.**

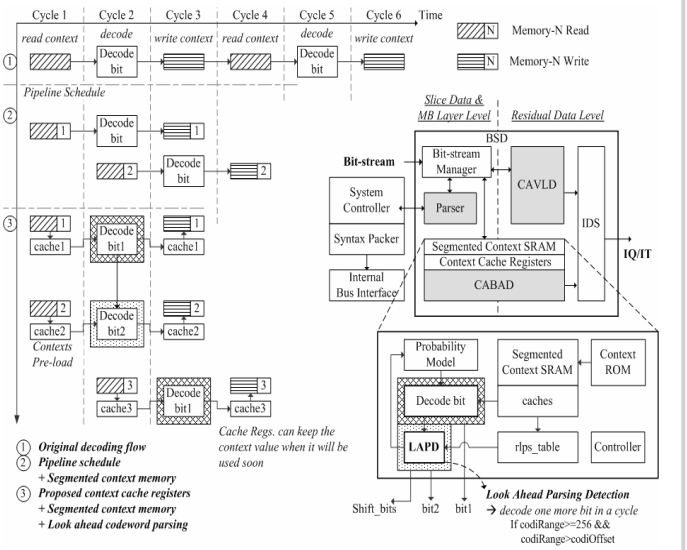



**Figure 22.3.2: Proposed optimization techniques on the CABAD operations.**

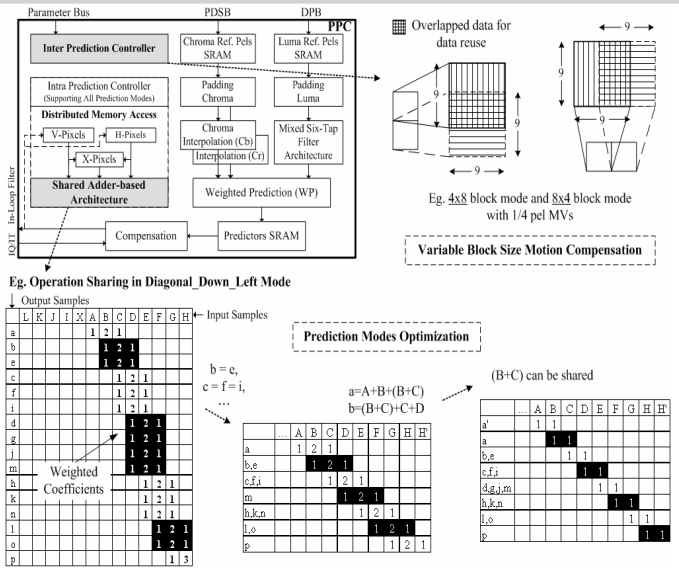
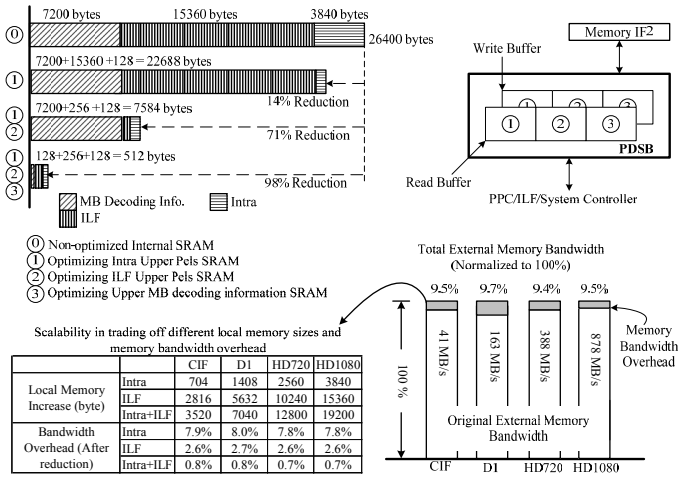



**Figure 22.3.3: Proposed optimization techniques on PPC operations.**

| Modules | Proposed Techniques | Improvement (Compared to) |
|---|---|---|
| CAVLD | Hierarchical Logic for Look-up Table (HLLT) | 39% gate-count reduction (Original realization*) |
| | Table partition | |
| | Zero-left Table Elimination by Arithmetic (ZTEA) | |
| CABAD | Segmented context memory | 57% cycle-count reduction (Original realization) |
| | Context cache registers | |
| | Look ahead codeword parsing | |
| PPC | Prediction modes optimization | 60% complexity reduction (Original realization) |
| | Mixed six-tap filter architecture | 27% gate-count reduction (ISCAS2005 [5]) |
| | Variable block size MC | 48% bandwidth reduction (Original realization) |
| ILF | Shared adder-based filter | 20% gate-count reduction (ISCAS2005 [4]) |
| | Horizontal-vertical interleaved raster-scan filtering order | 70% internal memory reduction (ISCAS2005 [4]) |
| | Progressive data allocation for MBAFF block filtering | 20% gate-count reduction (Original realization) |

*Note: The original realization refers to the realization of the H.264 modules without using the proposed techniques.*

**Figure 22.3.4: Proposed optimization techniques on H.264 video decoding process.**




Scalability in trading off different local memory sizes and memory bandwidth overhead

| | | CIF | D1 | HD720 | HD1080 |
|---|---|---|---|---|---|
| Local Memory Increase (byte) | Intra | 704 | 1408 | 2560 | 3840 |
| | ILF | 2816 | 5632 | 10240 | 15360 |
| | Intra+ILF | 3520 | 7040 | 12800 | 19200 |
| Bandwidth Overhead (After reduction) | Intra | 7.9% | 8.0% | 7.8% | 7.8% |
| | ILF | 2.6% | 2.7% | 2.6% | 2.6% |
| | Intra+ILF | 0.8% | 0.8% | 0.7% | 0.7% |

**Figure 22.3.5: Optimization in internal memory size through the proposed DMA-like PDSB.**

| | Proposed Design | ISCE '2004 [3] | ISCAS '2005 [4] | ISCAS '2005 [5] | ISCAS '2004 [6] |
|---|---|---|---|---|---|
| Specification | 1920x1088 @30fps | 2048x1024 @30fps | 2048x1024 @30fps | 1920x1088 @30fps | 1920x1088 @30fps |
| Profile | Baseline/Main | Baseline/Main | Baseline | Baseline | Baseline (Multi -standard) |
| Gate Count | 160K | 300K | 217K | 450K | 910K |
| Internal Memory | 4.5KB | 74KB | 10KB | >20KB | N/A |
| Clock Rate | 120MHz | 200MHz | 120MHz | 100MHz | 170MHz |
| Technology | 180nm (1.8V) | 130nm (1.2V) | 180nm (1.8V) | 180nm (1.8V) | 130nm (1.2V) |
| Power Consumption (180nm, 1.8V) | 320mW | --- | N/A | N/A | --- |
| Power Consumption (130nm, 1.2V) | 108mW (Post-sim) | 160mW | --- | --- | 554mW |

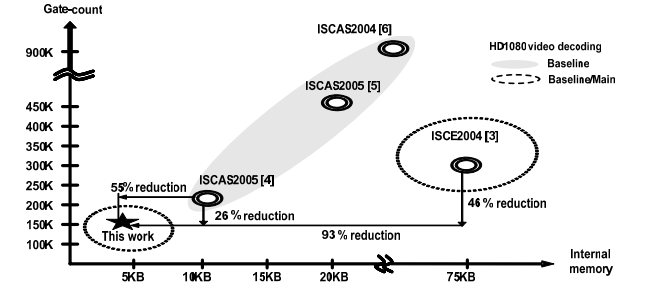



**Figure 22.3.6: Performance comparison with the existing H.264 video decoders.**

*Continued on Page 662*

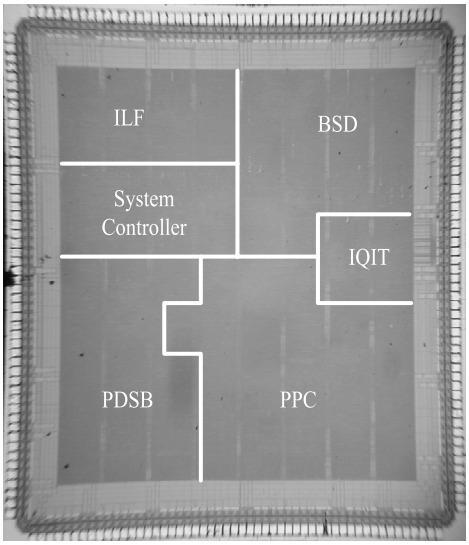



**Figure 22.3.7: Chip micrograph.**